

Санкт-Петербургский Государственный Университет
Математико-механический факультет

Кафедра параллельных алгоритмов

Дорожкина Алла Алексеевна

Распараллеливание методом декомпозиции области
численного решения уравнения диффузии

Бакалаврская работа

Научный руководитель:

Профессор Корнеев Вадим Глебович

Рецензент:

Профессор Демьянович Юрий Казимирович

Санкт-Петербург

2016

SAINT-PETERSBURG STATE UNIVERSITY

Department of Parallel Algorithms

Dorozhkina Alla

Parallel solution of diffusion equation by domain
decomposition method

Bachelor's Thesis

Scientific supervisor:

Professor Vadim Glebovich Korneev

Reviewer:

Professor Yuri Kazimirovich Dem'yanovich

Saint-Petersburg

2016

Оглавление

Введение.....	4
Постановка задачи	7
Решение	8
Программная реализация.....	10
Результаты	15
Заключение.....	23
Список литературы	24

Введение

Уравнение диффузии является частным видом дифференциального уравнения в частных производных. Решая уравнение диффузии, необходимо найти зависимость концентрации вещества (или объектов) от пространственных координат и времени. Если речь идет об общем случае, то данный коэффициент зависит от времени и пространственных координат.

В общем виде уравнение записывается следующим образом:

$$\frac{\partial \varphi(r, t)}{\partial t} = \nabla \cdot [D(\varphi, r) \nabla \varphi(r, t)],$$

где $\varphi(r, t)$ — плотность диффундирующего вещества в точке \mathbf{r} и во время t и $D(\varphi, r)$ — обобщённый диффузионный коэффициент для плотности φ в точке \mathbf{r} . ∇ - оператор набла.

Если коэффициент диффузии зависит от плотности — уравнение нелинейно, в противном случае — линейно.

Если D постоянное, то уравнение диффузии сводится к уравнению теплопроводности (линейному дифференциальному уравнению).

Уравнение диффузии было открыто в 1855 году известным немецким физиком и филологом Адольфом Фиком (1829 – 1901) по аналогии с уравнением Ж. Фурье для потока тепла [1]. Я. М. Гельфер в книге «История и методология термодинамики и статистической физики» (1981) отмечает: «Явление диффузии было впервые исследовано вюрцбургским ученым А.Фиком на примере соляных растворов. Фик путем тщательных исследований показал, что свободная диффузия соляных растворов происходит по законам, совершенно аналогичным законам распространения тепла в твердых телах». Уравнение диффузии бывает стационарным и нестационарным. Первое представляет собой параболическое

дифференциальное уравнение, которое описывает как распространяется растворимое вещество вследствие диффузии.

В случае одномерного диффузионного процесса с коэффициентом диффузии D уравнение приобретает вид:

$$\frac{\partial}{\partial t}c(x, t) = \frac{\partial}{\partial x}D\frac{\partial}{\partial x}c(x, t) + f(x, t).$$

А при постоянном D приобретает вид:

$$\frac{\partial}{\partial t}c(x, t) = D\frac{\partial^2}{\partial x^2}c(x, t) + f(x, t),$$

где $c(x, t)$ - концентрация диффундирующего вещества, а $f(x, t)$ - функция, описывающая источники вещества

В трехмерном случае уравнение выглядит следующим образом

$$\frac{\partial}{\partial t}c(\vec{r}, t) = (\nabla, D\nabla c(\vec{r}, t)) + f(\vec{r}, t),$$

где $\nabla = (\partial_x, \partial_y, \partial_z)$ - оператор набла, а $(,)$ - скалярное произведение.

Оно также может быть записано как

$$\partial_t c = \text{div} (D \text{grad } c) + f,$$

а при постоянном D имеет вид:

$$\frac{\partial}{\partial t}c(\vec{r}, t) = D\Delta c(\vec{r}, t) + f(\vec{r}, t),$$

где $\Delta = \nabla^2 = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$ - оператор Лапласа

Стационарное же уравнение относится к классу эллиптических уравнений. Данное уравнение рассматривается в случае задачи по нахождению установившегося распределения плотностью. Это означает, что можно исключить члены уравнения, которые связаны со временем в

нестационарном уравнении. Общий вид стационарного уравнения таким образом:

$$-(\nabla, D\nabla c(\vec{r})) = f(\vec{r}).$$

При D , не зависящем от \vec{r} , стационарное уравнение диффузии становится уравнением Пуассона $\Delta\varphi = f$, (данное уравнение неоднородное, описывает электростатическое поле, стационарное поле температуры, поле давления, поле потенциала скорости в гидродинамике) или уравнение Лапласа (однородное, т.е. при $f = 0$)

$$\Delta c(\vec{r}) = -\frac{f(\vec{r})}{D},$$

$$\Delta c(\vec{r}) = 0.$$

Численное решение уравнения Пуассона является важным элементом многих задач вычислительной физики. Так, например, при решении уравнений гидродинамики несжимаемой жидкости решение уравнения Пуассона требует значительных вычислительных усилий, особенно для расчетов с большим числом ячеек (от сотен тысяч и выше). [2] Метод декомпозиции является одним из методов быстрого решения данного уравнения. [3]

Постановка задачи

Необходимо решить уравнение Пуассона – для области $\Omega = \Omega_1 \cup \Omega_2$, являющейся объединением двух подобластей с непустым пересечением (Рис.1). На всей границе $\delta\Omega$ устанавливается однородное краевое условие Дирихле $u|_{\delta\Omega} = 0$. Провести распараллеливание полученного решения. Сравнить время выполнения прямой формы и параллельной.

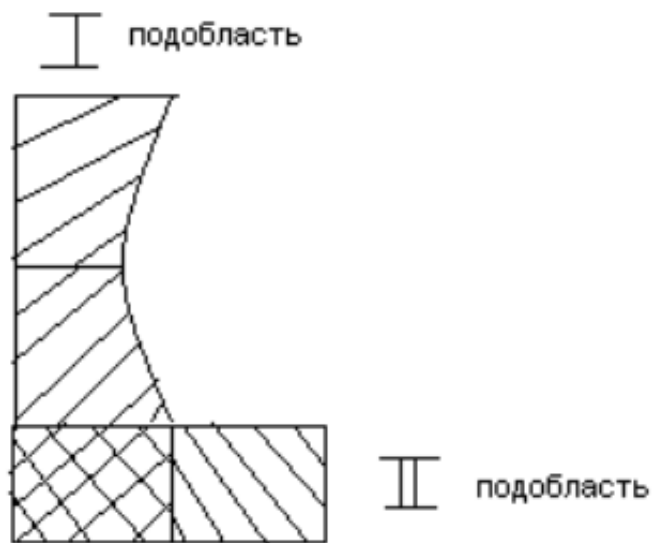


Рис. 1. Расчетная область

Решение

Проведем построение триангуляций области $\Omega = \Omega_1 \cup \Omega_2$ [4], обозначаемые T_h посредством таких четырехугольных сеток, что сетка на каждом участке области Ω_k топологически эквивалентна квадратной сетке с шагом h для прямоугольника, который обозначим через Ω_k^* . За сеточный параметр h принимаем среднее расстояние между двумя соседними узлами расчетной сетки МКЭ [5] с четырехугольными или треугольными конечными элементами первого порядка, имеющими прямолинейные стороны.

Пусть $K_h U_h = F_h$ – система алгебраических уравнений МКЭ для рассматриваемой задачи и $K_{h,k}$ – блок на диагонали матрицы K_h , отвечающий задаче Дирихле на Ω_k . Предобуславливатель [6] для матрицы K_h обозначаемый здесь B_h определяется через его обратный по формуле

$$B_h^{-1} = B_{h,1}^+ + B_{h,2}^+$$

где $B_{h,k}$ – прпредобуславливатель для блока $K_{h,k}$.

Псевдообратные матрицы для матриц $B_{h,k}$ имеют вид:

$$B_{h,1}^+ = \begin{pmatrix} B_{h,1}^{-1} & 0 \\ 0 & 0 \end{pmatrix}, \quad B_{h,2}^+ = \begin{pmatrix} 0 & 0 \\ 0 & B_{h,2}^{-1} \end{pmatrix}.$$

В качестве предобуславливателей для области декомпозиции выбирается матрица $B_{h,k} = \Delta_{h,k}$, где $\Delta_{h,k}$ пятиточечная матрица жесткости МКЭ для уравнения Лапласа в прямоугольнике на регулярной квадратной сетке шага h . Данные матрицы спектрально эквивалентны матрицам $K_{h,k}$.

Для решения задачи применяется предобусловленный метод сопряженных градиентов (PCG) [7] с предобуславливателем B_h .

Уравнение для итераций:

$$U^{k+1} = U^k + \delta B^{-1}(-KU + F)$$

Вектор решения уравнения Пуассона состоит из двух частей, соответствующих I и II области. Индексы значений для первой и второй области перекрываются, так как перекрываются сами области.

$$U^{(k+1)} = U_I^{(k+1)} + U_{II}^{(k+1)}$$

Получаем систему уравнений для нахождения решений для первой и второй подобласти итерационным методом.

$$\begin{cases} U_I^{(k+1)} = U_I + \delta B_I^{-1}(-K_I U_I + F_I) \\ U_{II}^{(k+1)} = U_{II} + \delta B_{II}^{-1}(-K_{II} U_{II} + F_{II}) \end{cases}$$

Где $V_I = -K_I U_I + F_I$ и $V_{II} = -K_{II} U_{II} + F_{II}$ - невязки для I и II подобластей

Введем обозначения:

$$W_I = B^{-1}(-K_I U + F_I)$$

$$W_{II} = B^{-1}(-K_{II} U + F_{II})$$

Невязки будем находить из следующих уравнений:

$$B_I W_I = V_I$$

$$B_{II} W_{II} = V_{II}$$

Таким образом система уравнений принимает вид:

$$\begin{cases} U_I^{(k+1)} = U_I + \delta W_I V_I \\ U_{II}^{(k+1)} = U_{II} + \delta W_{II} V_{II} \end{cases}$$

Программная реализация

Программная реализация решения выполнена с использованием пакета MATLAB [8]. Структура реализации выбрана следующая: имеется главная функция *main*, которая отвечает за построение области и вывод решений. Все вычисления инкапсулированы во второй функции – *solver*, которая собственно и производит решение уравнения Пуассона. Главная функция: *main*, передает в функцию *solver* область и получает решения, которые отображает в виде графиков.

В качестве исходных данных в функции *main* задается область, на которой решается задача и строится регулярная сетка для данной области. После этого задается область, соответствующая исходной, состоящая из секторов, необходимая для вычисления матриц предобуславливателей.

После задания области функция *main* вызывает функцию *solver* передавая области в качестве параметров.

Функция *solver* действует по описанному далее алгоритму. Происходит запуск цикла, который на каждом шаге увеличивает число разбиений области. Внутри цикла имеется:

- 1) Счетчик времени обработки одной итерации цикла, т.е. подсчета времени потраченного на построение решения.
- 2) Блок, который позволяет программе выполнять расчет как в параллельной форме, так и прямой в зависимости от значения параметра *is_parallel*

```
if is_parallel
    parfor i = 1:3
        if i == 1
            dim = [1 2 3];
        elseif i == 2
            dim = [3 4];
        else
            dim = [1 2 3 4];
```

```

end
[ks{i}, fs{i}] = assempde(b, p, e, t, 1, 0, 1, [], dim);
end
else
for i = 1:3
if i == 1
dim = [1 2 3];
elseif i == 2
dim = [3 4];
else
dim = [1 2 3 4];
end
[ks{i}, fs{i}] = assempde(b, p, e, t, 1, 0, 1, [], dim);
end
end

```

В данных циклах рассматривается пересечение участков области 1, 2, 3 и участков 3 и 4 (см. рис. 2). Выбор участков можно изменить, задав в качестве подобласти Ω_1 , например, участки 1 и 2, а в качестве Ω_2 участки под номерами 2, 3 и 4.

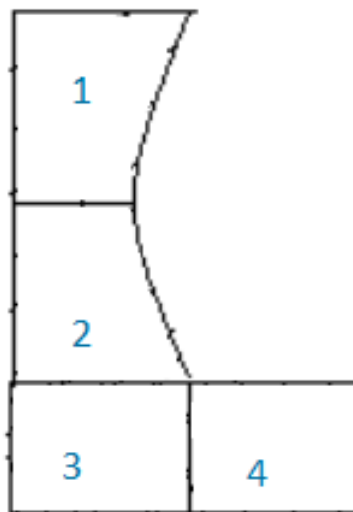


Рис. 2.

- 3) Вычисление матрицы жесткости и вектора F_I для уравнения $K_I U_I = F_I$, которое решается для первой подобласти Ω_1 , выделенной красным цветом на рис. 3

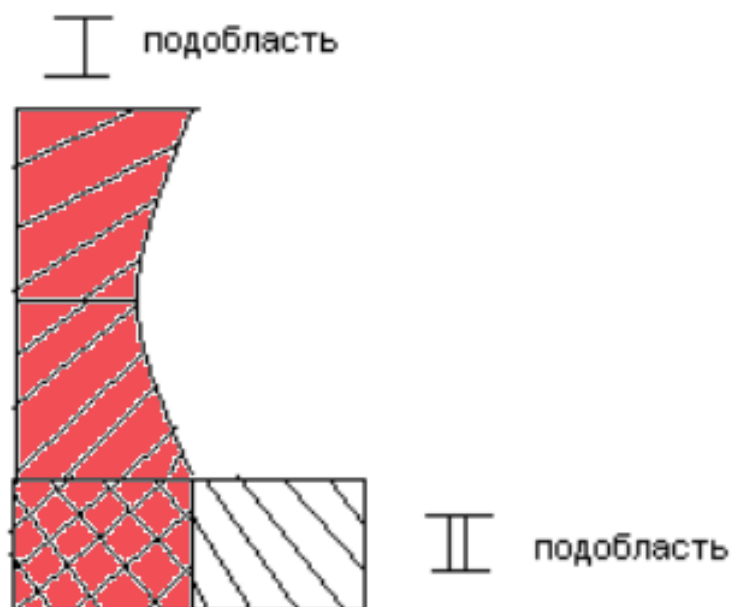


Рис. 3

- 4) Вычисление матрицы жесткости и вектора F_{II} для уравнения $K_{II} U_{II} = F_{II}$, которое решается для второй подобласти Ω_2 , выделенной синим цветом на рис. 4.

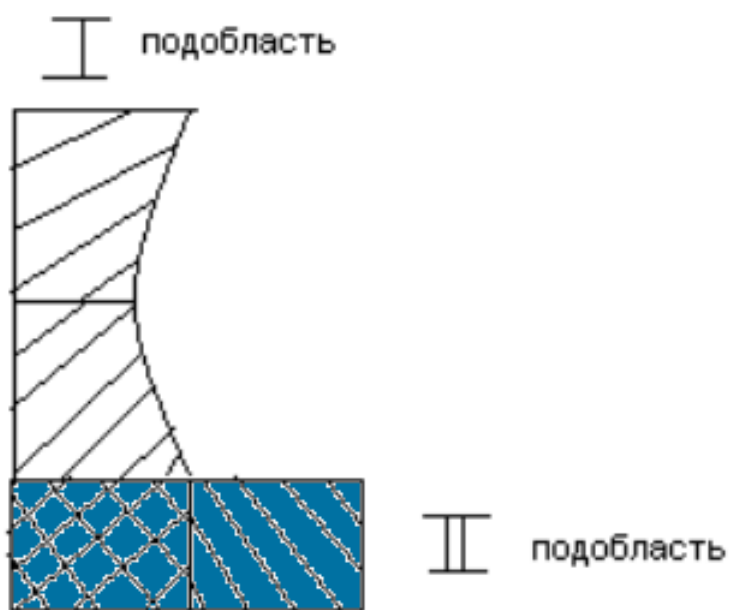


Рис. 4

- 5) Вычисление матрицы жесткости для предобуславливателя V_1 для области, представленной на рис. 3

- 6) Вычисление матрицы жесткости для предобуславливателя B_{II} для области, представленной на рис. 4
- 7) На каждом разбиении сетки строится итерационный процесс, который осуществляет решение уравнений. Задается первое приближение решения U для каждой подобласти. На каждой итерации происходит расчет нового значения U . Цикл работает до тех пор, пока норма разности решений на текущем и предыдущем шаге не станет меньше заданной погрешности. По достижении нормой этого числа – осуществляется выход из цикла и запоминается решение уравнения. Используются среднеквадратичная норма и норма, равная $\max |u(i)|$. В данном случае используется погрешность $1e-4$. Содержание цикла приведено ниже:

```
while d > 1e-4
    u_old = u;
    v1 = -K1 * u1 + F1;
    w1 = B1 \ v1;
    u1 = u1 + 1 * w1;
    u(i_up) = u1;
    u2 = u(i_down);
    v2 = -K2 * u2 + F2;
    w2 = B2 \ v2;
    u2 = u2 + 1 * w2;
    u(i_down) = u2;
    u1 = u(i_up);
    d = norm(u - u_old);

    iters(iter) = iters(iter) + 1;
end
```

После завершения работы вышеуказанного цикла в solver функция main строит следующие графики:

- 1) Набор решений исходного уравнения для различного количества точек и итераций
- 2) Зависимости числа итераций от количества вершин

3) Зависимость времени, потраченного на решение с разным количеством вершин для параллельной формы и прямой

Следующие важные моменты которые уменьшают время работы программы. Нахождение w из системы $Bw = v$, где B - предобуславливатель, а v – невязка лучше осуществлять с помощью быстрых методов. В настоящей реализации для нахождения используется двухслойная схема [9]. Для решения можно также применить многосеточный метод [10], который является более быстрым, но он сложнее в реализации. Результаты использование в данном случае, например, метода Гаусса, будут значительно хуже вышеприведенных методов.

Так же важным моментом, который существенно увеличивает производительность методов, является тот факт, что надо избегать заполнения и сохранения нулевых элементов матриц, а сохранять и умножать только ненулевые элементы.

Например, матрица B – матрица, получаемая из дискретизации оператора Лапласа и в ней только 5 отличных от нуля диагоналей. Поэтому двухслойная схема для такой матрицы будет работать быстро, если сохранить только ненулевые элементы (или эти 5 диагоналей).

Результаты

Ниже представлены графики решения уравнения для разного количества точек (рис. 5-12)

Number of nodes: 27

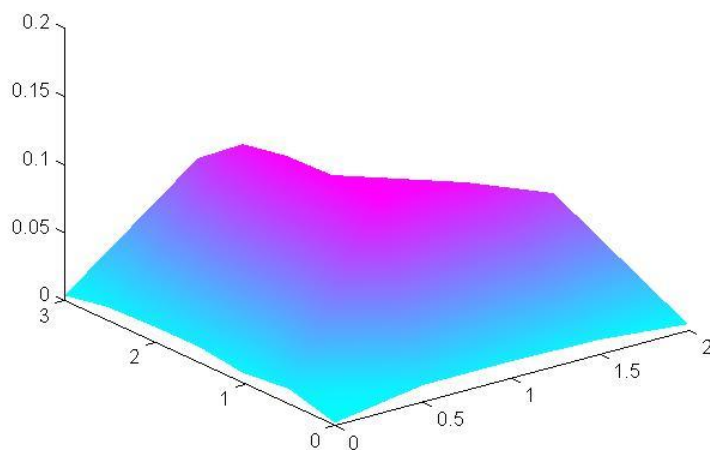


Рис. 5

Number of nodes: 85

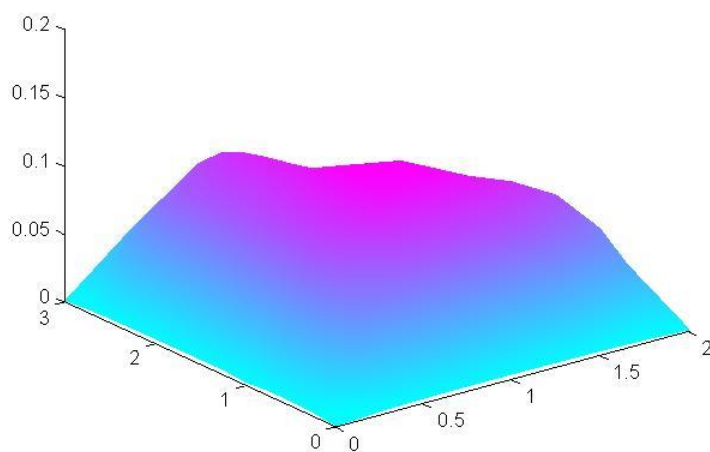


Рис. 6

Number of nodes: 1105

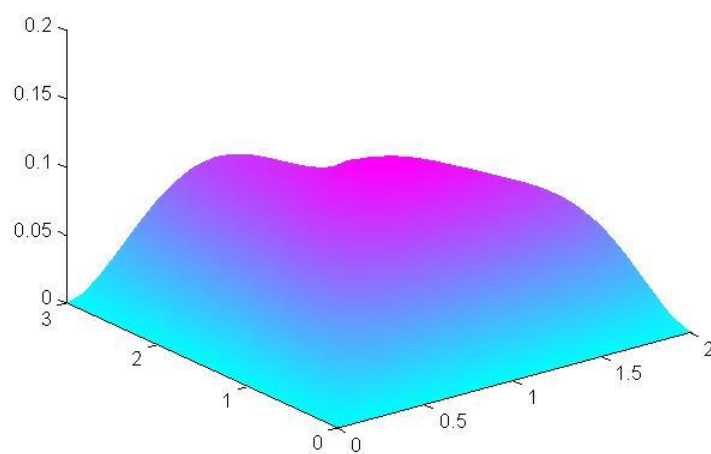


Рис. 7

Number of nodes: 4257

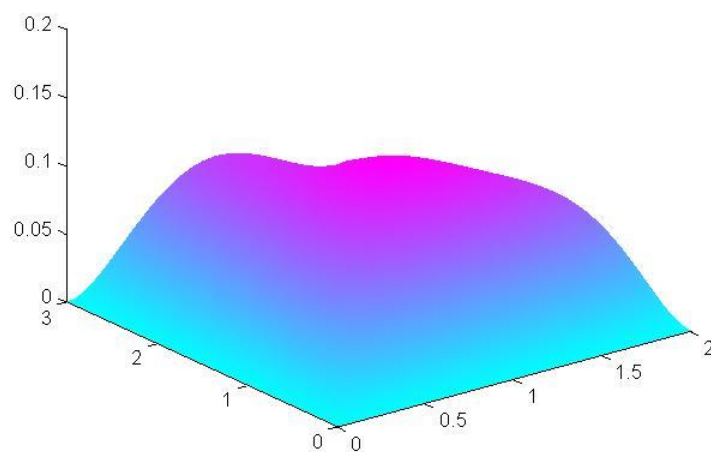


Рис. 8

Number of nodes: 297

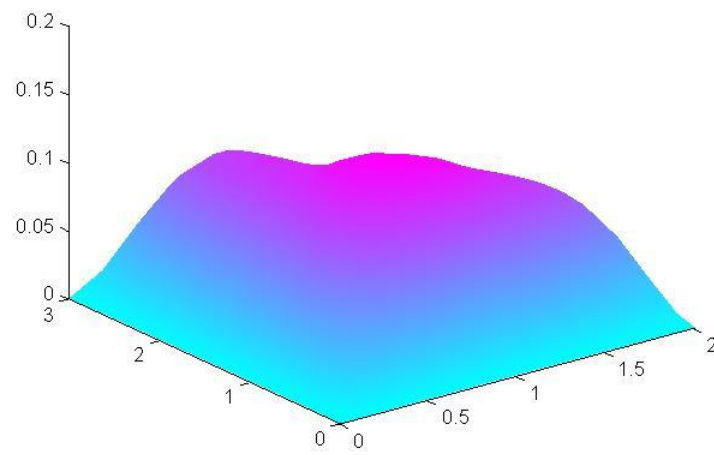


Рис. 9

Number of nodes: 16705

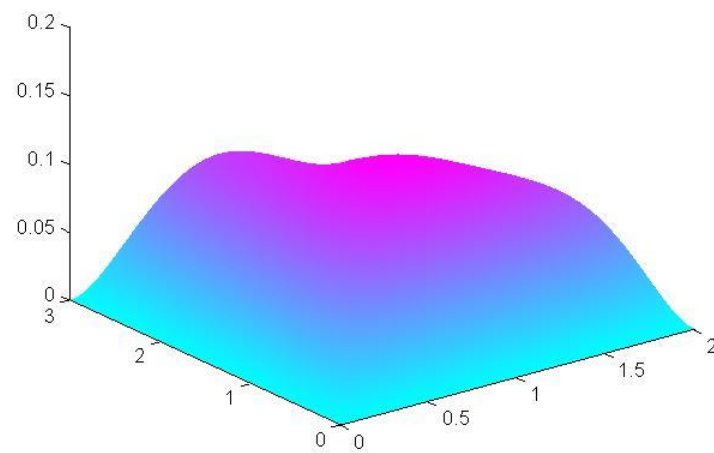


Рис. 10

Number of nodes: 66177

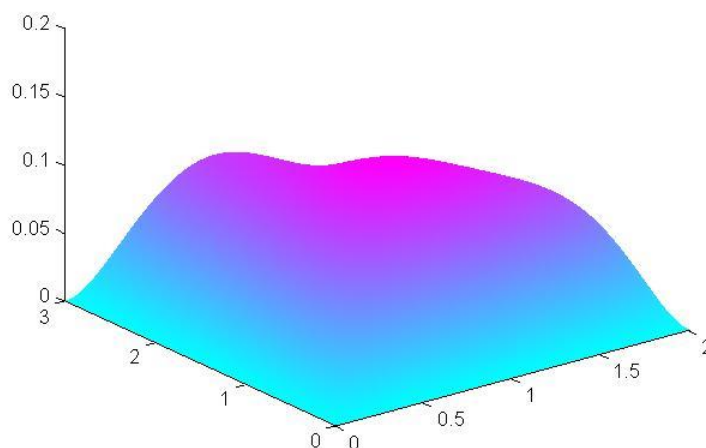


Рис. 11

Number of nodes: 263425

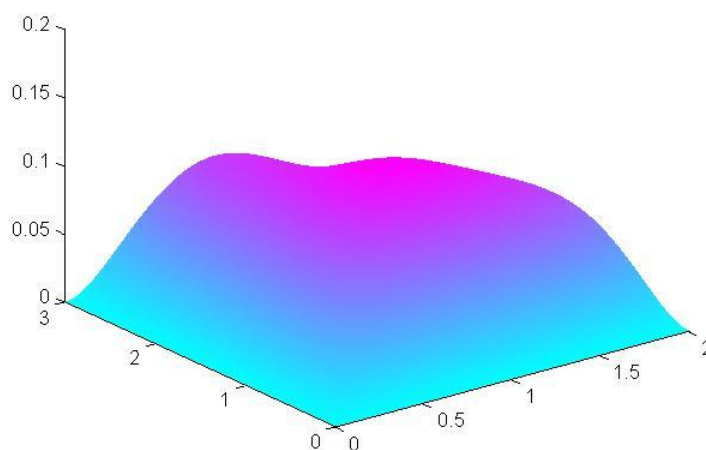


Рис 12.

На графиках, представленных ниже (рис. 13,14), показана зависимость количества вершин от времени в секундах. Пустыми кружочками обозначены точки для параллельного метода. Первоначальная инициализация занимает длительное время, в течение которого процессор создает потоки, что вносит дополнительные накладные расходы. В случае небольшого числа точек,

данные расходы занимают значительную часть от всего времени выполнения алгоритма. Поэтому при небольшом количестве точек расчетное время выполнения при использовании прямого метода несколько меньше чем при использовании параллельной формы. Для большого же числа точек выгоднее считать параллельной формой.

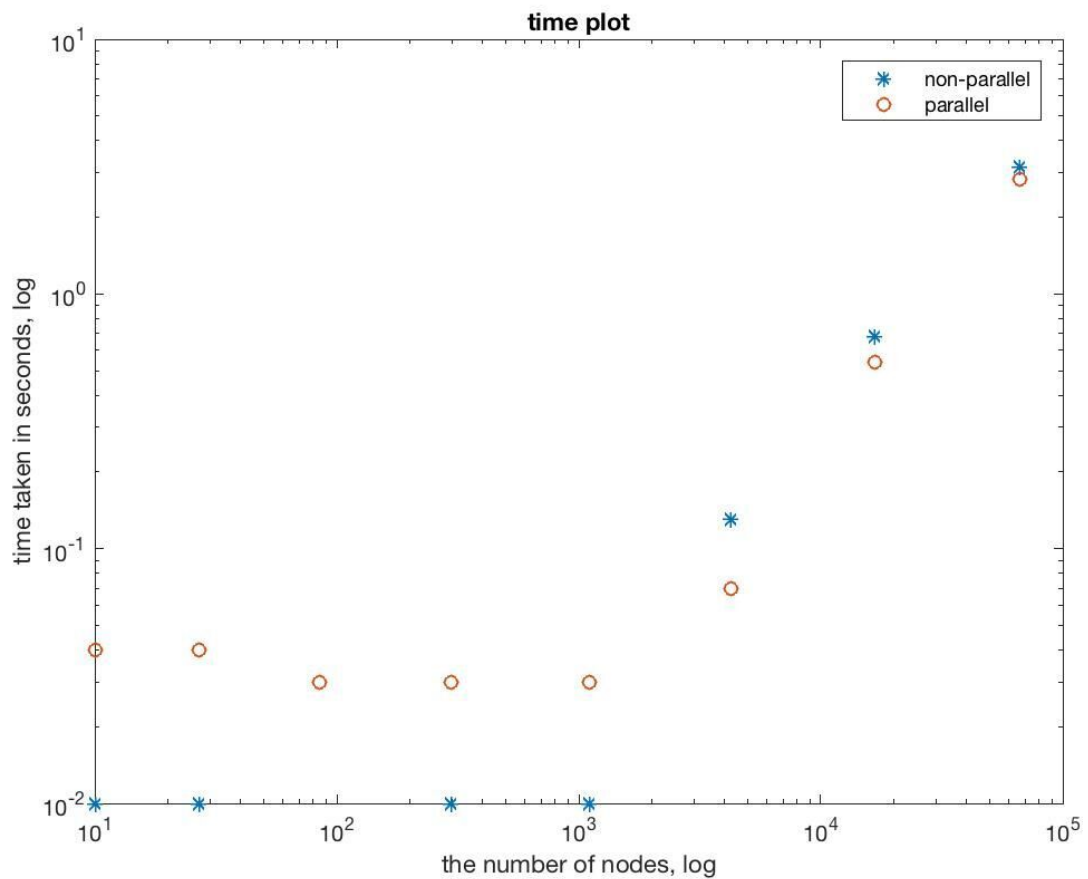


Рис 13.

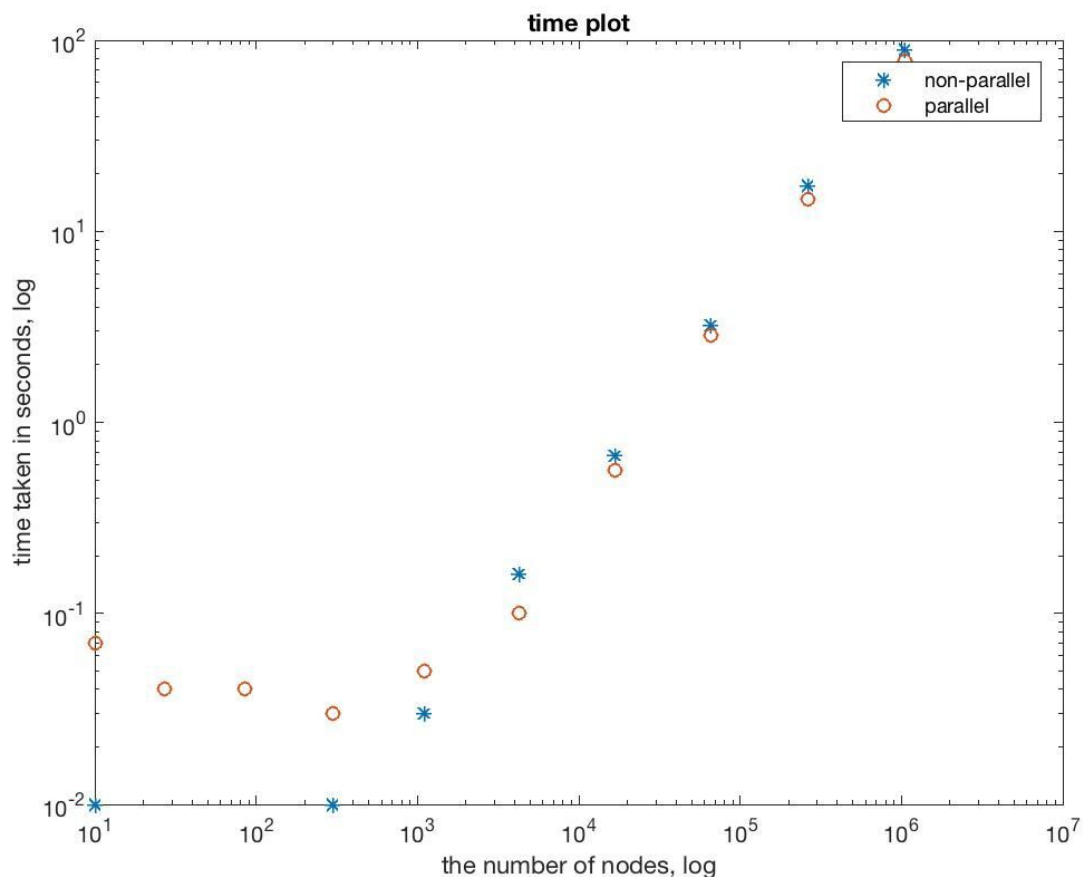


Рис. 14

Рассмотрев графики, можно понять, что время параллельной формы меньше, в данном случае незначительно, так как результаты представлены для работы на обыкновенном компьютере, имеющем только 2 ядра, следовательно, процессы распараллелены только на 2 потока.

В таблице 1 показана зависимость времени выполнения от количества итераций для параллельного и прямого метода. Время решения уравнения методом декомпозиции в обоих случаях достаточно мало. Для 8-ого построения, т. е. в том случае, когда количество точек равно 263425, время работы прямой формы равно 3.12 секундам, для параллельной 2.7

Итер.	Прямая форма (с.)	Параллельная форма (с.)
1	0	0.05
2	0.01	0.05
3	0.01	0.06
4	0	0.07
5	0.01	0.05
6	0.11	0.08
7	0.67	0.53
8	3.12	2.74

Таблица 1

Данные результаты можно сравнить с результатами, полученными с помощью метода Гаусса, который находит w , решая систему $Bw = v$, в которой B – предобуславливатель, а v – невязка. Уже на 5-ой итерации построение решения занимает почти 200 секунд. Зависимость времени от количества вершин для метода Гаусса показана на графике на рис. 15.

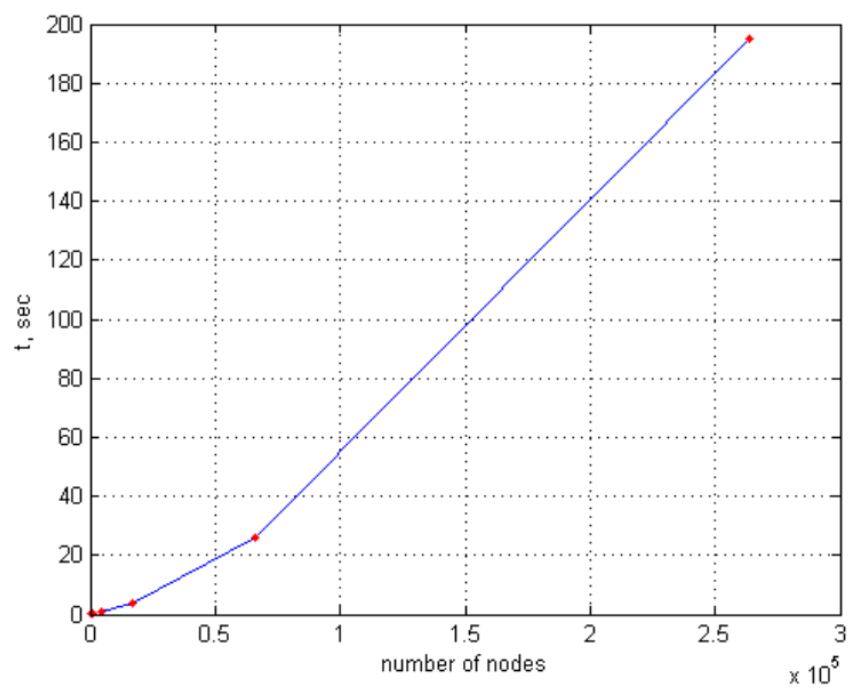


Рис. 15

Заключение

В ходе настоящей работы выполнена задача нахождения численного решения уравнения диффузии. Выявлено, что распараллеливание области методом декомпозиции является эффективным, что подтверждается полученными графиками решений и полученными результатами зависимости времени построения решения от количества точек. В соответствии с результатами, время, затраченное на решение уравнения Пуассона для достаточно большого количества точек (например, 263 тысячи, что является достаточно точным решением) является малым (для данного примера 2,7 секунды при параллельной форме и 3.1 секунды для прямой формы). Проведено сравнение применения метода декомпозиции с использованием быстрого метода решения системы уравнений и с использованием метода Гаусса. Второй способ менее эффективный. Для 4257 точек метод, разработанный в данной работе выполняет построение решения за 0.1 секунду при прямой форме работы программы и 0.5 при параллельной форме (разница с прямой формой такая, т к изначально происходит долгая инициализация всех данных, нужных для работы программы). Для метода декомпозиции в совокупности с методом Гаусса же время работы равняется 200 секундам.

Таким образом, применение метода декомпозиции с быстрым решением системы, отсутствием хранения неиспользуемых ячеек матриц и параллельным выполнением функций является эффективным в задачи нахождения области численного решения уравнения диффузии.

Список литературы

1. A. Fick, *Ueber Diffusion*, Pogg. Ann. Phys. Chem. 170 (4. Reihe 94), 59-86 (1855).
2. A Tutorial on Elliptic PDE Solvers and their Parallelization Craig C. Douglas, Gundolf Haase, Ulrich Langer
3. V. Korneev and U. Langer. Domain Decomposition Methods and Preconditioning. In: Encyclopedia of Computational Mechanics, V.1. E. Stein, R. de Borst and Th.J.R. Huges eds. 2004 John Wiley & Sons, Ltd.
4. А. А. Клячин, Равномерная триангуляция плоских областей, 1-6 (2011)
5. В.Г. Корнеев, С. Е. Енсен Эффективное предобуславливание методом декомпозиции области для р-версии с иерархическим базисом. I // Изв. вузов. Математика. - 1999. - №5. - С. 37-56.
6. В. Г. Корнеев, "Схемы метода конечных элементов высоких порядков точности для эллиптических уравнений второго порядка в трехмерных областях. 1979 год
7. Preconditioned conjugate gradients method.
<http://www.mathworks.com/help/matlab/ref/pcg.html>. 2016
8. MathWorks – Makers of MATLAB and Simulink
<http://www.mathworks.com/>
9. А.А. Самарский, Е.С. Николаев Методы решения сеточных уравнений Глава 6, Москва «НАУКА» Главная редакция физико-математической литературы 1978 год
10. В. В. Мареев, Е. Н. Станкова, Многосеточные методы. Введение в стандартные методы.: Издательство СПбГУ, 2012